Forgetting in Logic Programs under Strong Equivalence

Yisong Wang  
Department of Computer Science,  
Guizhou University, China

Yan Zhang and Yi Zhou  
Intelligent Systems Laboratory,  
University of Western Sydney, Australia

Mingyi Zhang  
Guizhou Academy of Sciences,  
Guiyang, China

Abstract

In this paper, we propose a semantic forgetting for arbitrary logic programs (or propositional theories) under answer set semantics, called HT-forgetting. The HT-forgetting preserves strong equivalence in the sense that strongly equivalent logic programs will remain strongly equivalent after forgetting the same set of atoms. The result of an HT-forgetting is always expressible by a logic program, and in particular, the result of an HT-forgetting in a Horn program is expressible in a Horn program; and a representation theorem shows that HT-forgetting can be precisely characterized by Zhang-Zhou’s four forgetting postulates under the logic of here-and-there. We also reveal underlying connections between HT-forgetting and classical forgetting, and provide complexity results for decision problems.

Introduction

Motivated from Lin and Reiter’s seminal work (Lin and Reiter 1994), the notion of forgetting in classical propositional and first-order logics has attracted extensive interests in KR community (Lang and Marquis 2010). In recent years, researchers have developed forgetting notions and theories in other non-classical logic systems from various perspectives, such as forgetting in logic programs (Zhang and Foo 2006; Eiter and Wang 2008; Wong 2009), forgetting in description logic (Wang et al. 2010; Lutz and Wolter 2011), and knowledge forgetting in modal logic (Zhang and Zhou 2009; Su et al. 2009; Liu and Wen 2011).

It is easy to see that for classical propositional logic, forgetting preserves logical equivalence. That is, logically equivalent formulas (theories) will remain logically equivalent after forgetting the same set of atoms. For logic programs, the issue of logical equivalence is rather complicated due to its different notions of “equivalence”: (weak) equivalence and strong equivalence. There have been several attempts to define the notion of forgetting in logic programs, but none of these approaches is fully satisfactory. Zhang and Foo (2006) first defined syntax oriented weak and strong forgetting notions for normal logic programs. But these forgetting notions preserve neither (weak) equivalence nor strong equivalence. Eiter and Wang (2008) then proposed a semantic forgetting for consistent disjunctive logic programs, which preserves equivalence but not strong equivalence. They specifically indicated the importance of preserving strong equivalence in logic programming forgetting and raised this issue as a future work. Wong (2009) proposed two forgetting operators for disjunctive logic programs. Although Wong’s forgetting indeed preserves strong equivalence, it may lose the intuition of weakening under various circumstances (see Related Work for details).

In addition to preserving strong equivalence, expressibility is another desired criterion for logic programming forgetting. Ideally we would expect that the result of forgetting some atoms from a logic program is still expressible by a logic program. Finally, we believe that as a way of weakening, forgetting in logic programs should obey some common intuitions shared by forgetting in classical logics. For instance, forgetting something from a logic program should lead to a weaker program in certain sense. On the other hand, such weakening should only be associated to the relevant information that has been forgotten. For this purpose, Zhang and Zhou (2009) proposed four forgetting postulates to formalize these common intuitions and showed that forgetting in classical propositional logic and modal logic S5 can be precisely captured by these postulates. Interestingly, none of previous forgetting notions in logic programs actually satisfies Zhang-Zhou’s postulates.

In summary, we consider the following criteria that a forgetting notion in logic program should meet:

- Expressibility. The result of forgetting in an arbitrary logic program should also be expressible via an arbitrary logic program;
- Preserving strong equivalence. Two strongly equivalent programs should remain strongly equivalence after forgetting the same set of atoms;
- Satisfying common intuitions of forgetting. Preferably, forgetting in logic programs should be semantically characterized by Zhang-Zhou’s four forgetting postulates.

In this paper we present a comprehensive study on forgetting in the context of arbitrary logic programs (propositional theories) under answer set semantics. In our approach, a program Π is viewed as a theory of the logic of here-and-there (or simply called HT logic), then forgetting a set V of
atoms from $\Pi$, is the theory consisting of the consequences of $\Pi$ in HT logic that mention no atoms from $V$. Our semantic forgetting meets all above criteria, and hence is of primary advantages comparing to previous logic program forgetting notions. We also investigate the relationship between the HT-forgetting and classical forgetting, and the relating computational complexity of HT-forgetting.

Preliminaries

Consider a propositional language $L$ over a finite set $A$ of propositional atoms (or is also called the signature of $L$). Formulas of $L$ are built from $L$’s signature $A$ and the 0-place connective $\bot$ ("false") using the binary connectives $\land, \lor$ and $\supset$. $\top$ ("true") is the shorthand of $\bot \lor \bot$, $\neg \varphi$ for $\varphi \supset \bot$, and $\psi \leftrightarrow \phi$ for $(\psi \lor \phi) \land (\phi \lor \psi)$. A theory is a set of propositional formulas. An interpretation is a set $I$ of atoms from $A$, where each atom of $A$ is viewed to be true if it is in $I$, and false otherwise. Then notions of model and satisfaction relation $\models$ are defined in a standard way. For two propositional formulas $\phi$ and $\psi$, $\phi \equiv \psi$ is used to denote $\phi \models \psi$ and $\psi \models \phi$.

Forgetting in propositional logic

Let $V$, $M_1$ and $M_2$ be sets of atoms. We say that $M_1$ and $M_2$ are $V$-identical, denoted by $M_1 \sim_{V} M_2$, if $M_1$ and $M_2$ agree on everything except possibly on $V$, i.e. $M_1 \setminus V = M_2 \setminus V$. Given a theory $\Sigma$, a theory, denoted as $\text{Forget}(\Sigma, V)$, is a result of forgetting $V$ in $\Sigma$, iff for any set $M$ of atoms, $M''$ is a model of $\Sigma$ iff there exists a model $M$ of $\Sigma$ such that $M \sim_{V} M''$. Alternatively, $\Sigma' \equiv \text{Forget}(\Sigma, V)$ iff

\[
\text{Mod}(\Sigma') = \{ M' \mid \exists M \models \Sigma \text{ s.t. } M \sim_{V} M' \},
\]

where $\text{Mod}(\Sigma)$ denotes the set of models of $\Sigma$. This is a semantic forgetting defined in (Lin and Reiter 1994). A syntactic counterpart of the semantic forgetting is given in (Lin 2001; Lang, Liberatore, and Marquis 2003) as mentioned in Introduction: $\text{Forget}(\Sigma, p) = \Sigma[p/T] \lor \Sigma[p/\bot]$, and $\text{Forget}(\Sigma, V \cup \{p\}) = \text{Forget}(\text{Forget}(\Sigma, p), V)$. These two definitions of forgetting are equivalent (cf. Corollary 5 of (Lang, Liberatore, and Marquis 2003)).

Answer sets for propositional theories

Given a formula $\psi$ and a set $X$ of atoms, the reduct $\psi^X$ of $\psi$ relative to $X$ is obtained from $\psi$ recursively via the following steps:

- for each atom $p$, if $X \models p$ then $p^X$ is $p$, otherwise it is $\bot$; $\bot^X = \bot$; and
- for each formula $\psi$ and $\phi$, if $X \models \psi \lor \phi$ then $(\psi \lor \phi)^X$ is $\psi^X \lor \phi^X$; otherwise it is $\bot$, where $\bot \in \{\lor, \land, \supset\}$.

The reduct $\Pi^X$ of a propositional theory $\Pi$ relative to $X$ is the set of $\psi^X$ with $\psi \in \Pi$. A set $X$ of atoms is an answer set (or stable model) of a set of propositional formulas $\Pi$ if $X$ is a minimal model (in terms of set inclusion) satisfying $\Pi^X$. Under this semantics, one should note that $\neg \varphi$ is not "equivalent to" $\varphi$, as $\neg \varphi$ has no answer set while $\{p\}$ is the unique answer set of $p$. Two theories $\Pi_1$ and $\Pi_2$ are strongly equivalent if $\Pi_1 \cup \Sigma$ and $\Pi_2 \cup \Sigma$ have the same answer sets for any theory $\Sigma$. Cabalar and Ferraris (2007) showed that a propositional theory (under answer set semantics) exactly captures a logic program with nested expressions (Lifschitz, Tang, and Turner 1999).

The logic of here-and-there

The syntax of HT logic is the same as classical propositional logic. An HT-interpretation is a pair $\langle H, T \rangle$ such that $H \subseteq T \subseteq A$. The satisfiability relation between an HT-interpretation $\langle H, T \rangle$ and a formula $\psi$, denoted by $\langle H, T \rangle \models \psi$, is recursively defined:

- $\langle H, T \rangle \models p$ if $p \in H$;
- $\langle H, T \rangle \not\models \bot$;
- $\langle H, T \rangle \models \psi_1 \lor \psi_2$ if $\langle H, T \rangle \models \psi_1$ or $\langle H, T \rangle \models \psi_2$;
- $\langle H, T \rangle \models \psi_1 \land \psi_2$ if $\langle H, T \rangle \models \psi_1$ and $\langle H, T \rangle \models \psi_2$;
- $\langle H, T \rangle \models \psi_1 \supset \psi_2$ if both (i) $T \models \psi_1 \supset \psi_2$, and (ii) $\langle H, T \rangle \models \psi_1$ implies $\langle H, T \rangle \models \psi_2$.

An HT-interpretation $\langle H, T \rangle$ is an HT-model of $\psi$ if $\langle H, T \rangle \models \psi$. An HT-model $(T, \Pi)$ of $\psi$ is an equilibrium model of $\psi$ if there is no $T'$ such that $T' \subseteq T$ and $\langle T', T \rangle \models \psi$ (Pearce 1996). The logic based on this semantics is called equilibrium logic. The satisfiability relation, HT-model and equilibrium model are extended to theories in a standard way. We denote Mod$_{ht}$($\Sigma$) the set of HT-models of theory $\Sigma$. In particular, if $\Sigma$ is a singleton $\{\psi\}$, we simply write it as Mod$_{ht}$($\psi$). Let $\Pi$ and $\Sigma$ be two theories. By $\Pi \equiv_{ht} \Sigma$, we mean that every HT-model of $\Pi$ is an HT-model of $\Sigma$, and by $\Pi \equiv_{ht} \Sigma$ we mean $\Pi \models_{ht} \Sigma$ and $\Sigma \models_{ht} \Pi$. In the latter case, we call $\Pi$ and $\Sigma$ are HT-equivalent. It has been shown that, a set $X$ of atoms is an answer set of a logic program $\Pi$ iff $(X, X)$ is an equilibrium model of $\Pi$. In addition, two logic programs $\Pi$ and $\Sigma$ are strongly equivalent iff $\Pi \equiv_{ht} \Sigma$ (Pearce, Tompits, and Woltraman 2001; Lifschitz, Pearce, and Valverde 2001; Ferraris 2011). The following proposition shows some basic properties of HT logic that we will need in our next study.

**Proposition 1** Let $\phi$ and $\psi$ be two formulas and $\langle X, Y \rangle$ an HT-interpretation.

(i) If $\langle X, Y \rangle \models \phi$ then $\langle Y, Y \rangle \models \phi$ (i.e., $Y \models \phi$),
(ii) $\langle X, Y \rangle \models \neg \phi$ iff $Y \models \neg \phi$,
(iii) $\langle X, Y \rangle \models \phi$ iff $X \models \phi^Y$,
(iv) If $\phi \equiv_{ht} \psi$, then $\phi \equiv \psi$.

$\neg \bot \equiv_{ht} \neg \bot \equiv_{ht} \top \equiv_{ht} \bot \equiv_{ht} \bot$.

HT-Forgetting in Logic Programs

Differently from previous approaches, our forgetting notion in logic programs will be based on the logic of here-and-there, which will lead to meet all criteria of forgetting we addressed earlier. Let $K_1 = \langle H_1, T_1 \rangle$ and $K_2 = \langle H_2, T_2 \rangle$ be two HT-interpretations and $V$ a set of atoms. $K_1$ and $K_2$ are said to be $V$-identical, denoted as $K_1 \sim_{V} K_2$, whenever $H_1 \sim_{V} H_2$ and $T_1 \sim_{V} T_2$. 

1In the rest of this paper, whenever there is no confusion, we may not explicitly mention the signature when we talk about formulas of $L$. 

Definition 1 Let $\psi$ be a formula and $V$ a set of atoms. A formula $\phi$ is called a result of HT-forgetting $V$ in $\psi$, iff the following condition holds:

\[
\text{Mod}_{ht}(\phi) = \{(H,T) \mid (H,T) \text{ is an HT-interpretation s.t. } \exists (X,Y) \in \text{Mod}_{ht}(\psi) \text{ and } (X,Y) \sim_{V} (H,T)\}.
\]

As we will see later, the forgetting result always exists and it is unique (up to strong equivalence), we will denote the forgetting result by $\text{Forget}_{ht}(\Pi,\psi)$ in what follows. From Definition 1, it is not difficult to see that HT-forgetting is independent of the order of forgetting atoms.

As HT-interpretations are related to a given signature, in what follows, we shall assume that the signature of a formula/theory is implicitly given by the atoms occurring in the formula/theory, unless explicitly stated otherwise.

Semantic Characterizations

Unlike propositional logic, HT logic does not hold a model characteristic property in general. That is, given a class of HT-interpretations, there may not exist a formula whose HT-models exactly correspond to those HT-interpretations. For example, let $M = \{\emptyset, \{a\}\}$, we can show that there is no formula that has a unique HT-model ($\emptyset, \{a\}$). To see this, suppose that there is a formula $\psi$ such that $\text{Mod}_{ht}(\psi) = M$, then we have $\{\{a\}, \{a\}\} \models_{ht} \psi$ by (i) of Proposition 1. But this is not the case. Hence, as a semantic forgetting notion, the study on the expressibility of HT-forgetting is important.

Expressibility

To begin with, we first introduce the notion of irrelevance which has a close connection to forgetting. A formula $\psi$ is HT-irrelevant to a set $V$ of atoms, denoted as $\text{IR}_{ht}(\psi, V)$, if there exists a formula $\phi$ mentioning no atoms from $V$ and $\psi \equiv_{ht} \phi$. For convenience, in the following, we will simply write a singleton set $\{a\}$ as $a$, and thus we may denote $\text{Forget}_{ht}(\psi, \{a\})$ as $\text{Forget}_{ht}(\psi, p)$, and $\text{IR}_{ht}(\psi, \{a\})$ as $\text{IR}_{ht}(\psi, p)$, etc.

Note that Definition 1 is semantic, which does not guarantee the existence of a formula $\phi$ such that $\text{Mod}_{ht}(\phi) = \text{Mod}_{ht}(\text{Forget}_{ht}(\psi, V))$. However, the following theorem states that such formula always exists.

Theorem 1 (Expressibility theorem) Let $\psi$ be a formula and $V$ a set of atoms. There exists a formula $\phi$ such that

\[
\text{Mod}_{ht}(\phi) = \{(H,T) \mid (H,T) \text{ is an HT-interpretation s.t. } \exists (X,Y) \in \text{Mod}_{ht}(\psi) \text{ and } (X,Y) \sim_{V} (H,T)\}.
\]

As one knows that disjunctive programs, positive programs, normal logic programs and Horn programs are four types of special cases of (arbitrary) logic programs under our setting. Then it is interesting to consider whether the expressibility result also holds for each of these special programs. For instance, we would like to know whether the result of forgetting in a disjunctive (positive, normal, and Horn) program is still expressible by a disjunctive (resp. positive, normal, and Horn) program. It turns out that the answer is negative for HT-forgetting in disjunctive, positive and normal programs.

Example 1 Consider the following normal logic program $\Pi$ over signature $\{p, q\}$:

\[
\neg p \supset q, \quad \neg q \supset p, \quad p \land q \supset \bot.
\]

$\Pi$ has two HT-models: $\langle\{p\}, \{p\}\rangle$ and $\langle\{q\}, \{q\}\rangle$. Then $\text{Mod}_{ht}(\text{Forget}_{ht}(\Pi,\psi))$ contains the HT-interpretations that are $\{p\}$-identical to one of the following HT-interpretations:

\[
\langle\emptyset, \emptyset\rangle, \quad \langle\{q\}, \{q\}\rangle,
\]

from which we conclude that $\text{Forget}_{ht}(\Pi, \psi) \equiv_{ht} q \lor \neg q$. It is shown that $q \lor \neg q$ cannot be expressed as a normal or disjunctive program in the sense that there is no normal or disjunctive program $\Pi$ which is strongly equivalent to $q \lor \neg q$ (Eiter, Tompits, and Woltran 2005).

Example 2 Let $\Pi$ be a positive logic program over signature $\{p, q, r\}$ as follows:

\[
p \lor q \lor r, \quad p \land q \supset r, \quad p \land r \supset q, \quad q \land r \supset p.
\]

By HT-forgetting $q$ from $\Pi$, we have $\text{Forget}_{ht}(\Pi, q) \equiv_{ht} (\neg r \lor p \lor \neg p) \land (\neg p \lor r \lor \neg r)$, which cannot be expressed by any positive logic program.

Theorem 2 (Horn program expressibility) Let $\Pi$ be a Horn logic program and $V$ a set of atoms. Then there exists a Horn logic program $\Pi'$ such that $\text{Forget}_{ht}(\Pi, V) \equiv_{ht} \Pi'$.

Strong equivalence and other properties

We can easily show that our HT-forgetting preserves strong equivalence. From Definition 1 and Theorem 1, the following result is obvious.

Proposition 2 Let $\psi$ and $\phi$ be two formulas and $V$ a set of atoms. If $\psi \equiv_{ht} \phi$, then $\text{Forget}_{ht}(\psi, V) \equiv_{ht} \text{Forget}_{ht}(\phi, V)$.

The property of preserving strong equivalence of HT-forgetting then immediately follows from the equivalent relationship between strong equivalence and HT-equivalence (Lifschitz, Pearce, and Valverde 2001; Pearce, Tompits, and Woltran 2009). The following proposition illustrates some essential properties of HT-forgetting.

Proposition 3 Let $\psi$ and $\phi$ be two formula and $V$ a set of atoms. Then the following results hold.

(i) $\psi$ has an HT-model if $\text{Forget}_{ht}(\psi, V)$ has.

(ii) $\psi \models_{ht} \text{Forget}_{ht}(\psi, V)$.

(iii) If $\psi \models_{ht} \phi$ then $\text{Forget}_{ht}(\psi, V) \models_{ht} \text{Forget}_{ht}(\phi, V)$.

(iv) $\text{Forget}_{ht}(\psi \lor \phi, V) \equiv_{ht} \text{Forget}_{ht}(\psi, V) \lor \text{Forget}_{ht}(\phi, V)$.

(v) $\text{Forget}_{ht}(\psi \land \phi, V) \models_{ht} \text{Forget}_{ht}(\psi, V) \land \text{Forget}_{ht}(\phi, V)$.

(vi) $\text{Forget}_{ht}(\psi \land \phi, V) \equiv_{ht} \text{Forget}_{ht}(\psi, V) \land \phi$ if $\text{IR}_{ht}(\phi, V)$.
Forgetting postulates

Zhang and Zhou (2009) proposed four forgetting postulates in their work on knowledge forgetting, and showed that their knowledge forgetting can be precisely characterized by these postulates. In the following, we show that HT-forgetting is exactly captured by these postulates, which we think is one major advantage over other logic program forgetting approaches.

The below proposition actually implies the uniform interpolation property of the logic of here-and-there.

**Proposition 4** Let \( \psi \) and \( \varphi \) be two formulas, \( V \) a set of atoms and \( \text{IR}(\varphi, V) \). Then we have

\[
\psi \models_{ht} \varphi \iff \text{Forget}_{ht}(\psi, V) \models_{ht} \varphi.
\]

Let \( \psi \) and \( \phi \) be two formulas and \( V \) a set of atoms. The following are Zhang-Zhou’s four postulates under the logic of here-and-there.

- **(W)** Weakening: \( \psi \models_{ht} \phi \).
- **(PP)** Positive persistence: if \( \text{IR}_{ht}(\xi, V) \) and \( \psi \models_{ht} \xi \) then \( \phi \models_{ht} \xi \).
- **(NP)** Negative persistence: if \( \text{IR}_{ht}(\xi, V) \) and \( \psi \not\models_{ht} \xi \) then \( \phi \not\models_{ht} \xi \).
- **(IR)** Irrelevance: \( \text{IR}_{ht}(\phi, V) \).

By specifying \( \phi \equiv_{ht} \text{Forget}_{ht}(\psi, V), (W), (PP), (NP) \) and \( (IR) \) are called postulates for HT-forgetting. Weakening \( (W) \) requires that forgetting results in weaker knowledge. The postulates of positive persistence \( (PP) \) and negative persistence \( (NP) \) simply state that forgetting a set of atoms should not affect those positive or negative information respectively that is irrelevant to this set of atoms. Finally, irrelevance \( (IR) \) means that after forgetting, the resulting knowledge should be irrelevant to those forgotten atoms.

**Theorem 3 (Representation theorem)** Let \( \psi \) and \( \phi \) be two formulas and \( V \) a set of atoms. Then the following statements are equivalent:

1. \( \phi \equiv_{ht} \text{Forget}_{ht}(\psi, V) \).
2. \( \phi \equiv_{ht} \{ \varphi \mid \psi \models_{ht} \varphi \text{ and } \text{IR}_{ht}(\varphi, V) \} \).
3. Postulates \( (W), (PP), (NP) \) and \( (IR) \) hold.

HT-forgetting and Classical Forgetting

It has been shown that strong equivalence of logic programs may be related to the equivalence of propositional logic (Pearce, Tompits, and Woltran 2001; Lin 2002). As the HT-forgetting does preserve strong equivalence, it is worth exploring further connections between HT-forgetting and the classical one.

**Proposition 5** Let \( \psi \) be a formula and \( V \) a set of atoms. The following results hold:

1. \( \text{Forget}_{ht}(\psi, V) \equiv \text{Forget}(\psi, V) \).
2. \( \text{Forget}(\neg \psi, V) \models_{ht} \text{Forget}_{ht}(\neg \psi, V) \).

The result (i) in Proposition 5 simply says that HT-forgetting and classical forgetting are equivalent under the classical propositional logic. From this result and Theorem 2, we immediately have the following corollary.

**Corollary 4** Let \( \Pi \) be a Horn program and \( V \) a set of atoms. Then \( \text{Forget}(\Pi, V) \) is expressible by a Horn program.

We also note that the converse of (ii) of Proposition 5 does not hold. That is, we usually do not have \( \text{Forget}_{ht}(\psi, V) \models_{ht} \text{Forget}(\neg \psi, V) \).

Proposition 6 provides a method of computing HT-forgetting in a Horn program through its corresponding classical forgetting, since we know that \( \text{Forget}(\Pi, V \cup \{p\}) \equiv \text{Forget}_{ht}(\Pi, V) \) and \( \text{Forget}(\psi, p) \equiv \psi[p/\top] \lor \psi[p/\bot] \) for any formula (program) \( \psi \).

**Proposition 6** Let \( \Pi \) be a Horn program, \( V \) a set of atoms and \( \Pi \equiv \text{Forget}(\Pi, V) \).

**Proposition 7** Let \( \psi \) and \( \phi \) be two formulas and \( V \) a set of atoms. The following results hold.

1. \( \phi \equiv_{ht} \text{Forget}(\psi, V) \iff \neg \phi \equiv_{ht} \text{Forget}_{ht}(\neg \psi, V) \).
2. \( \text{Forget}(\phi, V) \equiv \text{Forget}(\psi, V) \iff \text{Forget}_{ht}(\neg \phi, V) \equiv_{ht} \text{Forget}_{ht}(\neg \psi, V) \).

Computational Complexity

Given a formula \( \psi \) and a set \( V \) of atoms, from Theorem 3 (i.e. (ii) in the representation theorem), we can see that the result of the HT-forgetting \( \psi \) is \( \text{Forget}(\psi, V) \), can be generated by computing all logical consequences of \( \psi \) which are HT-irrelevant to \( V \). Indeed, since there is a sound and complete axiomatic system for the logic of here-and-there (Jongh and Hendriks 2003), this is feasible. Nevertheless, it is also observed that from a computational viewpoint, like the classical forgetting, the process of generating \( \text{Forget}_{ht}(\psi, V) \) would be expensive as shown by the below theorem.

**Theorem 5** Let \( \psi \) and \( \phi \) be two formulas and \( V \) a set of atoms. We have

1. deciding if \( \psi \equiv_{ht} \text{Forget}_{ht}(\phi, V) \) is \( \Pi^P_n \)-complete,
2. deciding if \( \psi \models_{ht} \text{Forget}_{ht}(\phi, V) \) is \( \Pi^P_n \)-complete,
3. deciding if \( \text{Forget}_{ht}(\psi, V) \models_{ht} \phi \) is \( \text{coNP} \)-complete.

Related Work

As mentioned in Introduction, several approaches of forgetting in logic programs have been developed earlier in the literature. While Zhang and Foo’s weak and strong forgetting (Zhang and Foo 2006), and Eiter and Wang’s semantic forgetting (Eiter and Wang 2008) do not preserve strong equivalence, Wong’s forgetting operators (Wong 2009) does. In the section we compare our HT-forgetting with the ones proposed by Wong.

Wong developed his forgetting for disjunctive logic programs. Differently from Zhang and Foo’s and Eiter and Wang’s approaches, Wong’s forgetting is defined based on the logic of here-and-there. In this sense, Wong’s approach
probably shares a common logic ground with HT-forgetting. Wong also defined two forgetting operators $F_S$ and $F_W$, which correspond to two series of program transformations. The interesting feature of Wong’s forgetting is that it preserves strong equivalence.

However, a major issue with Wong’s forgetting is that: on one hand, forgetting may cause unnecessary information loss; on the other hand, forgetting may also introduce extra information that we do not want, as illustrated by the following example.

**Example 3** Let us consider the normal logic program $\Pi$ consisting of:

\[
\begin{align*}
a & \leftarrow x, \\
y & \leftarrow a, \neg z, \\
q & \leftarrow \neg p, \\
p & \leftarrow q, \\
& \leftarrow p, q.
\end{align*}
\]

Then we have:

\[
\begin{align*}
F_S(\Pi, \{a, p\}) & \equiv_{ht} \{y \leftarrow x, \neg z\}, \\
F_W(\Pi, \{a, p\}) & \equiv_{ht} \{y \leftarrow x, \neg z, \rightarrow x, \rightarrow q\}, \\
\text{Forget}_{ht}(\Pi, \{a, p\}) & \equiv_{ht} \{y \leftarrow x, \neg z, q \leftarrow \neg \neg q\}.
\end{align*}
\]

Since $\Pi \models_{ht} \{q \leftarrow \neg \neg q\}$, which is irrelevant to atoms $a$ and $p$, it seems to us that forgetting $\{a, p\}$ from $\Pi$ should not affect this fact. But clearly $F_S(\Pi, \{a, p\}) \not\equiv_{ht} \{q \leftarrow \neg \neg q\}$. In this sense, we see $F_S$ has lost some information that we wish to keep.

On the other hand, from the fact that $\Pi \not\models_{ht} q$ but $F_W(\Pi, \{a, p\}) \models_{ht} q$, it appears that $F_W$ may introduce unnecessary information, which indeed conflicts our intuition of program weakening via forgetting.

\[
\square
\]

**Concluding Remarks**

In the paper we proposed a semantic forgetting in arbitrary logic program which preserves strong equivalence and also satisfies other important semantic properties that previous forgetting approaches do not have.

Some related issues remain for the future work. Forgetting in logic programs has demonstrated its applications in conflict resolution and knowledge base update (Zhang and Foo 2006; Eiter and Wang 2008). We believe that our HT-forgetting can be used to develop a general framework for knowledge bases merging and update where each knowledge base is represented as an arbitrary logic program.

**Acknowledgement**

We appreciate the reviewers for their detailed comments. Yisong Wang and Mingyi Zhang are partially supported by the Natural Science Foundation of China under grant 60963009, while Yisong is also partially supported by Open Funds of the State Key Laboratory of Computer Science of Chinese Academy of Science under grant SYSKF1106.

**References**


